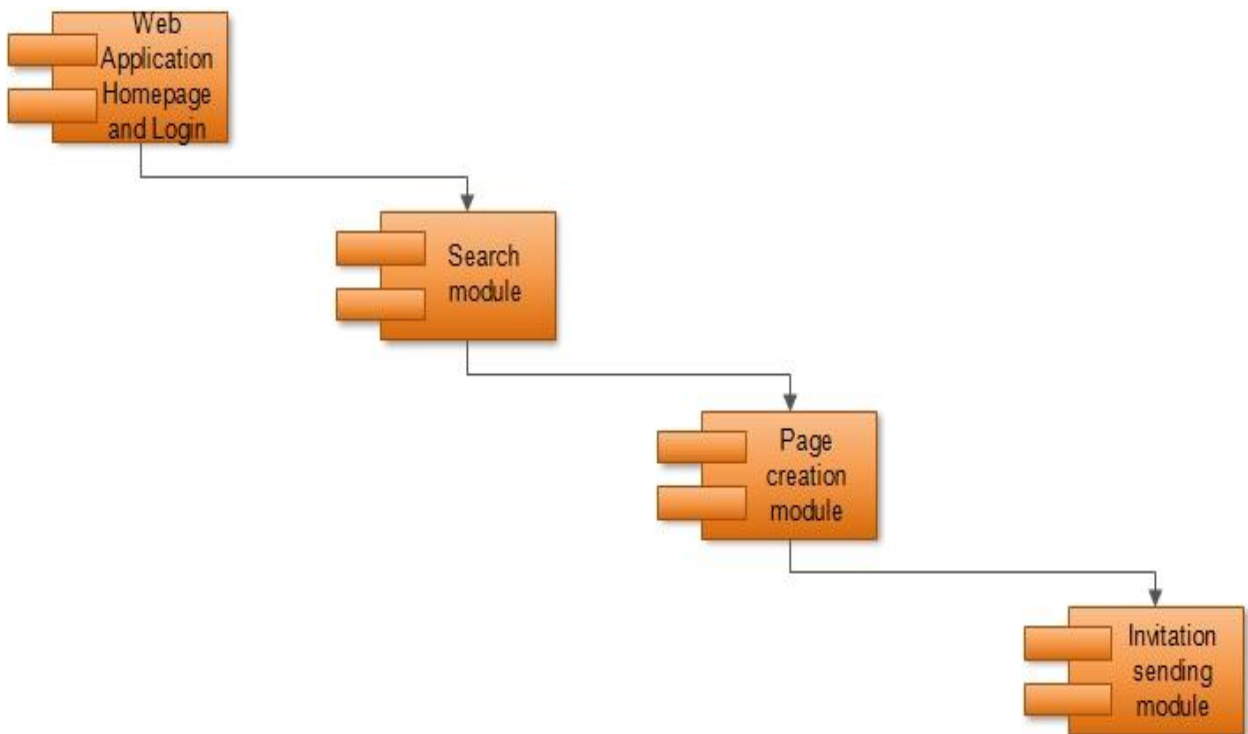


Product Design

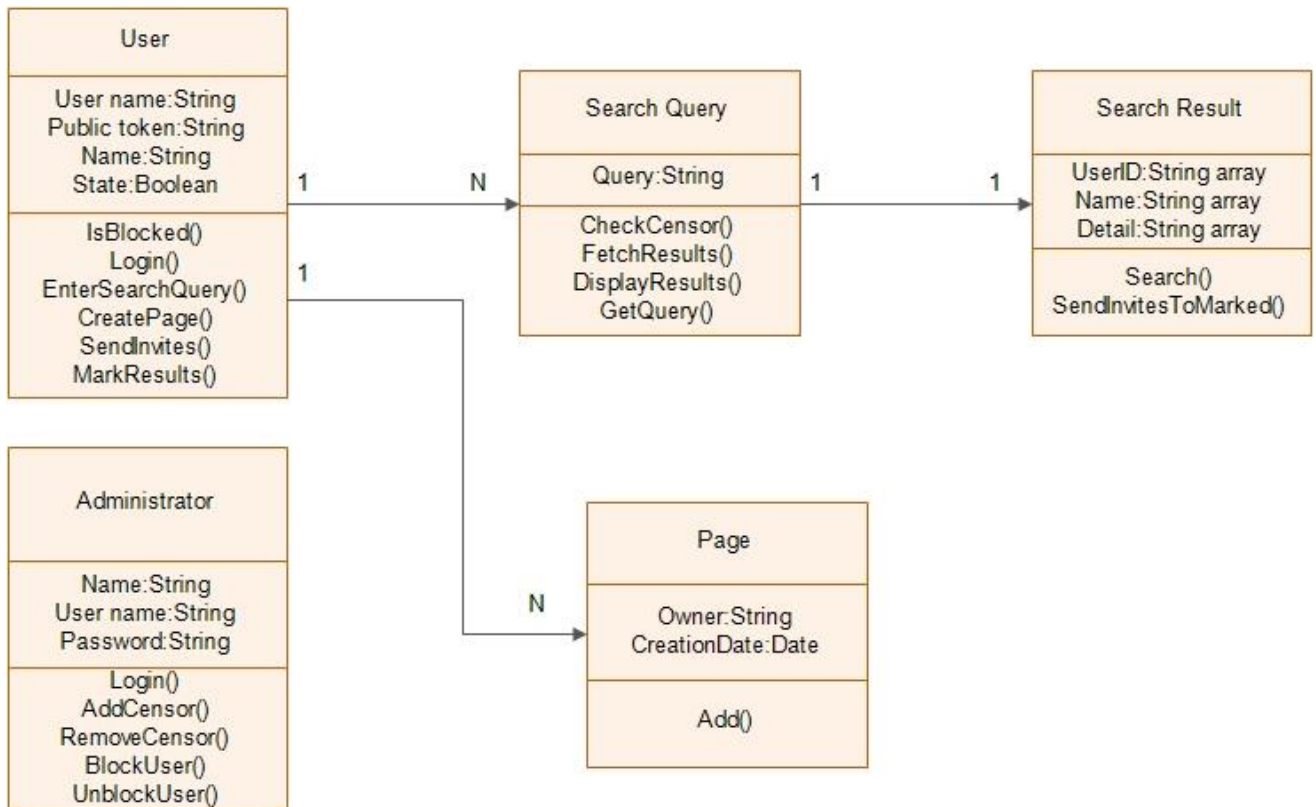
Team number	27
Project Title	<i>Purposeful facebook resource/application</i>
Document	SSAD Project Concept Document
Creation date	<i>November 5, 2013</i>
Created By	<i>Mahak Gambhir</i>
Client	<i>Dr. Kamalakar Karlapalem, Dean Academics</i>

Architectural Model



Web Application Homepage and login	<p>Component State</p> <ul style="list-style-type: none"> • The login state of the user, and if logged in, with which social networking site. • Whether he has authorized the application for that account. <p>Component Behavior</p> <ul style="list-style-type: none"> • Login of the user with the desired social networking site. • Authorization of the application by the user. • Redirecting to the search page on login.
Search module	<p>Component State</p> <ul style="list-style-type: none"> • The past searches by the user. • The list of censored searches. • Whether the entered query is censored or not. • Whether the user wishes to continue. <p>Component Behavior</p> <ul style="list-style-type: none"> • Reporting the user if the query is censored. • Showing only the new results if the query has been processed earlier. • Showing all the results otherwise. • Continue to page creation if user wishes to continue
Page Creation module	<p>Component State</p> <ul style="list-style-type: none"> • Whether the user is allowed to create a page by the social networking website. • Whether the search operation was successful or not. <p>Component Behavior</p> <ul style="list-style-type: none"> • If the user is prohibited to make a page, then he is reported and the process aborted. • Move on to invite people to the page on successful page creation
Invitation Sending module	<p>Component State</p> <ul style="list-style-type: none"> • Whether the page was created or not. • The search results of the query. • The list of people selected from the search results. <p>Component Behavior</p> <ul style="list-style-type: none"> • Wait for user to select the desired invitees. • Send all these people a request to like the created page.

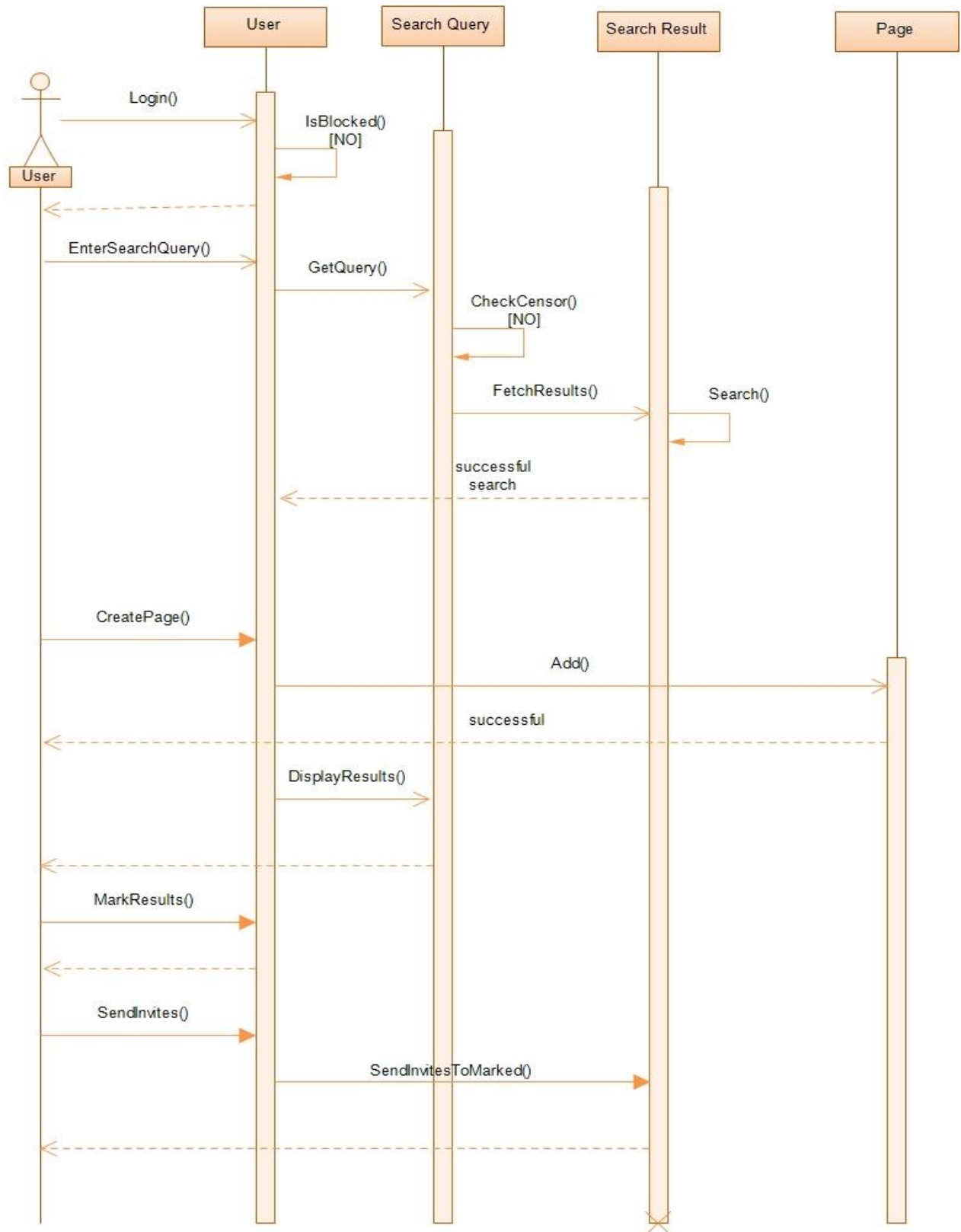
Class Diagram



NOTE:

All the functions of the administrator refer to the database and hence he has no association with any of the other classes.

Sequence Diagram



Design Patterns

The design pattern is essentially the “Chain of Responsibility” behavioral pattern whereby all the requests made by the user are effectively passed through a sieve-like structure, verified for various points at successive stages until they are finally ‘handled’ by the last class.

Design Rationale

The design has been made keeping in mind that the user has to be minimally involved in the entire process without any inconvenience. This calls for maintaining an interactive state for the user while the processes go on internally. In the light of the above facts, the user has been given only utility functions and doing so, the implementation details have been abstracted affectively.

The classes store only the relevant data and the other data requirements are met through a database access.

Code smells like a “God Object/Class” and “Data Classes” have been avoided by adhering to the design pattern of “Chain of Responsibility” and yet “feature/neighbour’s envy” has also been eschewed by giving each class the set of functionalities that it will need frequently and can satisfactorily handle.

The impending presence of “circular dependencies” in the code has been saved shrewdly by employing the much popular “Observer Pattern”, which is an integral element of the MVC architecture used by web2py, our development platform.